



La variabilité due à la sensibilité au contexte dans les processus téléologiques

Rebecca Deneckere, Elena Kornyshova

► To cite this version:

Rebecca Deneckere, Elena Kornyshova. La variabilité due à la sensibilité au contexte dans les processus téléologiques. INFORSID, 2010, Marseille, France. pp.161-176. hal-00703632

HAL Id: hal-00703632

<https://hal.science/hal-00703632>

Submitted on 4 Jun 2012

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

La variabilité due à la sensibilité au contexte dans les processus téléologiques

Rébecca Deneckère, Elena Kornyshova

*Centre de Recherche en Informatique
Université Paris 1 Panthéon-Sorbonne, 90 rue de Tolbiac, 75013 Paris, France
{rebecca.deneckere, elena.kornyshova}@univ-paris1.fr*

RÉSUMÉ. *La variabilité s'est avérée être un concept central dans différents domaines d'ingénierie comme la fabrication, le développement de logiciels, etc. afin de développer des solutions pouvant être facilement adaptables à différents contextes organisationnels et/ou à différents types de clients pour un meilleur coût. Les processus téléologiques ont, par nature, un haut niveau de variabilité. Notre position est que l'intégration de la « sensibilité au contexte » dans ce type de processus permettra d'augmenter cette variabilité inhérente aux processus téléologiques avec l'ajout de critères spécifiques. Nous proposons une typologie d'indicateurs ainsi qu'un processus d'utilisation de ces critères et nous illustrons notre propos avec le modèle de processus intentionnel MAP.*

ABSTRACT. *Variability has proved to be a central concept in different engineering domains, manufacturing, software development etc. in order to develop solutions that can be easily adapted to different organizational settings and different sets of customer at a low price. Teleological processes have, by nature, a high level of variability. Our position is that the integration of "context-awareness" in that kind of processes increases their intrinsic variability with specific criteria. We propose an indicator typology with an usage process and illustrate it with the MAP intentional process model.*

MOTS-CLES *Orienté-intention, Modèle de Processus, Variabilité, Sensibilité au contexte*

KEYWORDS *Intention-oriented, Process Model, Variability, Context-awareness*

1 Introduction

La variabilité s'est avérée être un concept central dans différents domaines d'ingénierie comme la fabrication, le développement de logiciels, etc. afin de développer des solutions pouvant être facilement adaptables à différents contextes organisationnels et/ou à différents types de clients pour un meilleur coût.

Les processus téléologiques sont des processus prenant en compte les intentions couplées aux résultats que l'utilisateur souhaite obtenir. D'un point de vue téléologique, tous les processus d'ingénierie de systèmes d'information (SI) contiennent des étapes de deux sortes: la prise de décision et l'exécution de la décision. Ces deux étapes alternent l'une avec l'autre. D'une part, la réalité téléologique de l'existence de processus exige des décisions. D'autre part, les décisions résultent de la nécessité de satisfaire des contraintes d'ordre pratique comme la qualité, les coûts ou les performances (Ruhe, 2003). Bien que ces aspects soient importants, les arguments pour prendre les décisions finales sont assez pauvres. Les choix sont effectués de manière intuitive et hasardeuse (Ruhe, 2003) (Ngo-The *et al.*, 2005). La nature téléologique des processus se retrouve donc essentiellement dans les processus dits intentionnels. Ces processus sont flexibles par nature puisqu'ils prennent en compte les différents points de variations donnant lieu à un choix entre plusieurs intentions (Rolland, 2007). On peut donc dire qu'ils ont une variabilité intrinsèque à leur définition. Cette variabilité peut être augmentée avec la prise en compte de la variabilité due à la sensibilité au contexte.

Notre position est que l'intégration de la sensibilité au contexte dans ce type de processus permettra d'augmenter la variabilité inhérente aux processus téléologiques avec l'ajout de critères spécifiques. A cette fin, nous proposons une typologie d'indicateurs dont les valeurs permettent une prise de décision plus formelle. Nous illustrons notre proposition avec l'introduction de cette typologie dans le modèle de processus téléologique MAP.

Cet article est organisé comme suit. La section 2 détaille les notions de processus téléologiques et de variabilité. La section 3 approfondit le concept de sensibilité au contexte. Notre proposition est décrite dans la section 4 et illustrée dans la section 5. La section 6 ouvre une discussion sur la notion de ligne de processus et nous concluons dans la section 7.

2 Les Processus Téléologiques et la Variabilité

L'ingénierie des processus est considérée comme essentielle (Rolland, 1998) à la fois par la communauté de l'ingénierie des logiciels et par la communauté de l'ingénierie des systèmes. Par exemple, (Dowson *et al.*, 1993) (Amenise *et al.*, 1993)

(Jarke *et al.*, 1994) démontrent que l'amélioration d'un processus de développement conduira à une meilleure productivité et à une meilleure qualité des systèmes.

Les processus de développement traditionnels ont été conçus comme des suites d'étapes, chaque étape dominée par une activité particulière organisée en différentes séquences (par exemple, « l'analyse », « la conception », « le développement », « les tests » (Royce, 1970) (Boehm, 1988). Plus tard, ces séquences d'activités ont été remplacées par des valeurs et des pratiques (Beck, 2005) (Abrahamsson, 2002). (Ralph, 2008) met en évidence le fait que, dans la pratique, le développeur du logiciel est confronté à une myriade de choix méthodologiques, puisqu'il peut choisir d'exécuter une activité entièrement ou partiellement, combiner des aspects de deux ou plusieurs activités différentes, ou en ignorer une autre. Les décisions prises au cours de l'exécution conduisent l'ingénieur à adopter un processus qui est unique et complètement adapté à la situation en cours.

Toutefois, les points de décision ne sont souvent abordés que de manière partielle. La plupart des processus connus n'ont pas les moyens de prendre en compte la prise de décisions inhérente à tout processus de développement de systèmes d'information pour le développement. Selon Dowson (Dowson, 1988), les modèles de processus peuvent être classés en trois groupes de modèles appelés orientés-activités, orientés-produit et orientés-décision. (Rolland, 1998) raffine cette classification avec un quatrième groupe appelé orientés-contexte. Les processus orientés-décision et orientés-contexte essaient de prendre en compte la prise de décision en intégrant les concepts d'objectifs et d'arguments. Les modèles orientés-décision sont non seulement capables d'expliquer comment le processus est effectué, mais aussi pourquoi il est effectué (Rolland, 1998) (Ralph, 2008). Les modèles de processus orientés-contexte couplent la notion de contexte de décision à la décision elle-même (Grosz *et al.*, 1997) (Rolland *et al.*, 2000). Cela rend la notion de contexte (le couplage d'une situation et d'une décision) un concept central des processus de modélisation. Ces deux types de modèles de processus peuvent être définis comme des *processus téléologiques*.

Un processus téléologique est un processus qui prend en compte le comportement téléologique d'une exécution d'un processus (comportement attaché à la notion de but). Il décrit les intentions (buts, objectifs) associées à un résultat que le concepteur veut obtenir. (Taylor, 1964) propose une classification de comportement comme étant soit axé sur les buts (téléologique) soit axé sur la réponse aux stimuli. (Malcom, 1967) définit un comportement téléologique comme suit : lorsqu'un comportement spécifique est nécessaire pour satisfaire un objectif, alors ce comportement se produit. En fait, il est nécessaire d'étudier la possibilité que les règles changent au cours du processus de développement. (Cayla, 2008) souligne le lien entre cette possibilité de changement et les principes téléologiques de Veblen (Veblen, 1898) qui intègrent les intentions de l'acteur dans les processus. Ce travail définit un processus téléologique comme « un équilibre dynamique dont les règles sont déterminées de façon endogène, comme une relation entre le comportement du système et de ses intentions ». Un tel processus téléologique peut exprimer les changements dans une trajectoire de processus. Par exemple, même si l'ingénieur a décidé d'un chemin général pour l'exécution de son

processus, il a l'occasion de modifier ce chemin (au cours de l'exécution), en prenant en compte les informations et les connaissances qu'il a recueilli de la situation en cours.

(Van de Ven, 1995) définit les processus téléologiques comme décrivant le comportement des acteurs faisant des actions pour atteindre un but ou un objectif (un état final envisagé). (Ralph, 2008) souligne que cette théorie est compatible avec le développement de logiciels de la façon suivante : les agents téléologiques sont l'équipe de développement logiciel et le chef de projet; l'état final représente le produit logiciel, l'équipe de développement effectue des actions (codage, tests ...) qui correspondent aux étapes téléologiques et le chef de projet surveille les performances. La notion de but est également essentielle pour toute organisation. Le SI est créé car il y a plusieurs objectifs à atteindre et ses fonctions et propriétés sont définies par les objectifs de l'organisation que le système vise à remplir (Loucopoulos *et al.*, 1995). Par exemple, les objectifs des processus sont envisagés dans des approches comme KAOS et I*. Ces approches définissent la notion d'option alternative existant dans les processus d'ingénierie des exigences. Les intentions alternatives sont évaluées pour améliorer ces processus (Van Lamsweerde, 2009).

Ces processus téléologiques intègrent, par le biais des décisions à prendre pour réaliser les intentions, une certaine variabilité. Au fil du temps, la *Variabilité* en génie logiciel est devenu de plus en plus important. Dans les premiers temps, le SI permettait de réaliser les objectifs d'une seule organisation et d'un ensemble assez simple de clients, alors que, de nos jours, le SI doit être conçu dans une perspective plus large, pour répondre aux objectifs de plusieurs organisations, être adaptable à des situations et des clients différents (Rolland, 2007). La variabilité est la capacité à être sujet à la variation. La notion de variabilité du logiciel est donc définie comme la capacité d'un système logiciel à être modifié, adapté ou configuré dans un certain contexte (Van Gurp, 2000). Alors que la communauté du logiciel considère la variabilité comme un problème de conception et se concentre sur les questions d'implémentation (Svanberg, 2001) (Bosh *et al.*, 2001) (Bachmann *et al.*, 2001), nous estimons, comme (Halmans, 2003), que l'expression de la variabilité au niveau des buts est essentielle pour satisfaire la nature multi-objectifs de ces nouveaux SI, intégrant la variabilité et capables de s'auto-adapter à la situation en cours.

(Liaskos *et al.*, 2007) établit que la modélisation de la variabilité est utile tant pour l'acquisition de variabilité – pour découvrir des points de variation – que pour l'analyse de la variabilité – pour évaluer l'applicabilité de chaque variant identifié dans un contexte et une situation donnés. Le grand nombre de caractéristiques observées dans les systèmes informatiques modernes, ainsi que la difficulté à comprendre comment les détails techniques des décisions affectent les intentions des parties prenantes, conduit à la nécessité d'explorer et d'analyser la variabilité à un niveau supérieur d'abstraction. Plusieurs typologies de variabilité ont été proposées dans cette optique. Par exemple, Salifu *et al.* (Salifu, 2007) utilisent les descriptions de problèmes pour représenter et analyser la variabilité dans les logiciels sensibles au contexte ; ce travail identifie le lien entre les besoins et le contexte comme une étape fondamentale pour concevoir des systèmes sensibles au contexte. Ils proposent quatre dimensions à considérer pendant

l'établissement de la variabilité contextuelle et les préoccupations liées aux exigences de qualité, les phénomènes physiques, l'application d'un processus de prise de décision aux besoins de qualité et l'identification des catégories connues de problèmes. (Bashmann *et al.*, 2001) distingue plusieurs sources de variation telles que les fonctions, les données, les contrôles de flux, la technologie, les objectifs qualité et l'environnement. Dans (America *et al.*, 2003), la classification décrit les vues clients, application, fonctionnelle, conceptuelle et développement. Dans (Kuloor *et al.*, 2003), l'identification des dimensions de variabilité des besoins est considérée comme un problème de séparation des besoins spécifiques du problème étudié alors que, dans (Halmans *et al.*, 2003), la variabilité essentielle (la variabilité d'un point de vue utilisateur) se distingue de la variabilité technique (problèmes d'implémentation).

La variabilité d'un processus se doit de prendre en compte le contexte du projet sur lequel il s'applique. Pour cela, on parle de la *sensibilité au contexte*.

3 La sensibilité au contexte

(Dey *et al.*, 2001) définit la notion de *contexte* par toute information pouvant être utilisée pour caractériser la situation d'une entité (personne, objet physique ou informatique). Et, plus généralement, tout élément pouvant influencer le comportement d'une application. (Rey *et al.*, 2002) propose plusieurs axes de définitions du contexte: il n'y a pas de contexte sans contexte (i.e. le contexte doit se définir en fonction d'une finalité), le contexte est un espace d'information qui sert l'interprétation (i.e. la capture du contexte n'est pas l'objectif mais les données capturées doivent servir un but), le contexte est un espace d'information partagé par plusieurs acteurs (i.e. utilisateur et système), le contexte est un espace d'information infini et évolutif (i.e. il se construit au cours du temps).

Les modèles de contexte sont pluridisciplinaires (Bradley *et al.*, 2005). La recherche linguistique est concernée par l'analyse du contexte d'utilisation des signes (mots) dans un langage. (Bunt, 1997) définit cinq types de contexte pour les aspects communication, qui sont respectivement : linguistique (lié au matériel linguistique), sémantique (lié à la description du domaine), physique (lié à la description de l'environnement dans lequel l'action ou l'interaction se produit), social (lié à la situation interactive qui se produit entre acteurs) et cognitif (lié aux intentions des participants, leur évolutions dépendant de la perception, la production, l'évaluation et l'exécution).

Le contexte est également formalisé selon des modèles mathématiques. Par exemple, (Coutaz *et al.*, 2002) propose un modèle cumulatif où le contexte (Ctx) est une agrégation temporaire de situations. Une situation est un descripteur d'état pour un utilisateur (U) exécutant une tâche (T) à un certain moment (t). Le modèle est décrit par la formule suivante:

$$Ctx(U, T, t) = \bigcup_{n=1}^m (Situation(U, T, t_n))$$

Le terme '*Context-awareness*' (sensibilité au contexte) vient des travaux sur l'informatique pervasive, ou *informatique ubiquitaire*. Ces systèmes utilisent les modifications de l'environnement dans les systèmes informatiques. Même si c'est un terme d'origine informatique, il a également été appliqué à la théorie des entreprises dans le domaine de la gestion des processus métiers (Rosemann *et al.*, 2006). C'est dans (Schilit *et al.*, 1994) que l'on trouve la première apparition de ce terme qui ne concerne alors que des variables comme la localisation, l'identité des personnes proches et des objets ainsi que les changements dans ces objets. Plusieurs définitions existent, comme (Dey *et al.*, 2001) pour qui un système est sensible au contexte s'il utilise le contexte pour offrir des informations ou des services pertinents pour l'utilisateur alors que (Salber *et al.*, 1998) définit la sensibilité au contexte comme étant la meilleure capacité d'un système à agir en temps réel avec des données provenant du contexte. Plus en rapport avec notre étude, les modèles de contexte sont également proposés pour la réingénierie des processus métiers (Bessai *et al.*, 2008), l'informatique (Bradley *et al.*, 2005), la sélection de services (Kirsh Pinheiro *et al.*, 2008) et la prise de décision en situation militaire (Rosen *et al.*, 2008), (Drury *et al.*, 2008). Dans ces derniers cas, le modèle de contexte est perçu comme une manière d'analyser une *situation* donnée afin de guider une exécution de tâches. Ainsi, les modèles de contexte sont principalement utilisés pour résoudre le problème de manque de flexibilité et d'adaptabilité au sein des processus.

4 Variabilité des processus téléologiques due à la sensibilité au contexte

Nous avons mentionné que les processus téléologiques contiennent des étapes de prise de décision. Ces décisions peuvent porter sur les modèles à utiliser, actions à effectuer, scénarios possibles etc. Ce sont notamment les décisions prises au cours de l'exécution des processus téléologiques qui permettent de les adapter au contexte et, par conséquent, d'assurer leur variabilité. Chaque fois qu'un ingénieur doit choisir entre plusieurs possibilités pour continuer son processus, il est confronté à un problème de prise de décision.

Un problème de prise de décision est défini par la présence d'au moins deux alternatives. Le concept d'alternative désigne l'objet de la décision. Les alternatives doivent être identifiées clairement. Ces alternatives sont comparées entre elles selon un ou plusieurs critères. Les critères représentent des informations qui permettent d'évaluer et de comparer les alternatives. Ils peuvent être de nature différente : les caractéristiques intrinsèques des objets de décision, avis des parties prenantes, conséquences potentielles des alternatives etc. Le problème de prise de décision comporte également la problématique de décision qui représente la solution cible et peut représenter : un choix (sélection d'un sous-ensemble d'alternatives potentielles ou une alternative, le cas échéant) ; un rangement (ordonnance des alternatives potentielles) ou bien un tri (affectation des alternatives potentielles à un ensemble de classes prédéfinies). De cette manière, le problème de prise de décision contient trois éléments de base : la problématique, les alternatives et les critères (Roy, 2005).

Dans le cadre de notre approche chaque étape de prise de décision dans les processus téléologiques est considérée en tant qu'un problème de prise de décision. Plus précisément, la problématique est le choix, les alternatives sont des variantes potentiellement disponibles dans un processus sur chacun des points de variation et les critères sont des indicateurs de contexte.

Typologie d'indicateurs. Les processus téléologiques contiennent des points de variation où l'ingénieur doit prendre une décision entre plusieurs alternatives. Il se retrouve donc confronté à un problème de choix. L'évaluation de ce problème, et sa résolution, peuvent être envisagées de plusieurs manières. Par exemple, la méthode I* préconise l'utilisation de la notion d'exigences non fonctionnelles en utilisant les 'soft goals' identifiés comme des critères d'évaluation. Les alternatives contribuent alors à différents degrés de satisfaction de ces buts (van Lamswerde, 2009). Dans notre approche, pour résoudre le problème de prise de décision de notre ingénieur, nous proposons d'utiliser le contexte du projet. Ce contexte est composé d'un ensemble d'indicateurs permettant de le caractériser (figure 1).

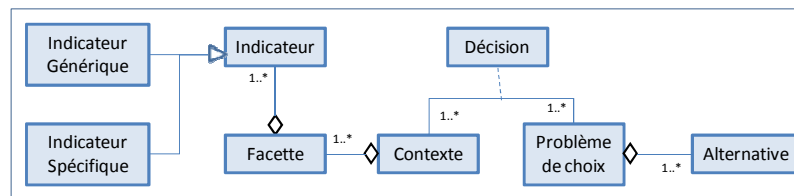


Figure 1. Sensibilité au contexte des processus téléologiques

Afin d'établir cette typologie, nous avons défini les caractéristiques couvrant les aspects essentiels de l'ingénierie des projets de SI, sur la base de (Mirbel *et al.*, 2006), (Van Slooten *et al.*, 1996) et (Kornyshova *et al.*, 2007). Nous distinguons deux types d'indicateurs : génériques et spécifiques. Les premiers sont communs à la plupart des modèles de processus alors que les autres varient d'un modèle de processus à l'autre. Cette typologie est décrite plus en détails dans (Deneckère *et al.*, 2010) et (Kornyshova *et al.*, 2010).

Processus d'utilisation. Le processus d'utilisation de cette typologie est appelé IG ('Indicator-based Guidance' ou *Guidage basé sur les Indicateurs*) et est illustré à la figure 2. Ce processus est représenté par le modèle MAP utilisé dans cet article pour illustrer la proposition et qui est expliqué dans la section suivante. Il s'effectue à deux niveaux différents : au niveau modèle et au niveau instance.

Au niveau du modèle de processus, l'ingénieur de méthode doit tout d'abord intégrer le concept général d'indicateur dans le modèle de processus initial pour déterminer quels sont les concepts pour lesquels le fait de rattacher des caractéristiques de contexte a un sens. De plus, une fois ces concepts déterminés, il lui faudra définir quels sont les différents indicateurs s'appliquant à chacun d'entre eux (*par expertise du modèle*). Il est également possible de définir des indicateurs spécifiques à chaque modèle de processus, qu'il faut également rattacher à des concepts précis (*par*

identification d'indicateurs spécifiques). L'ingénieur de méthode a également la possibilité d'effectuer des ajustements sur les indicateurs utilisés (*Par expertise*).

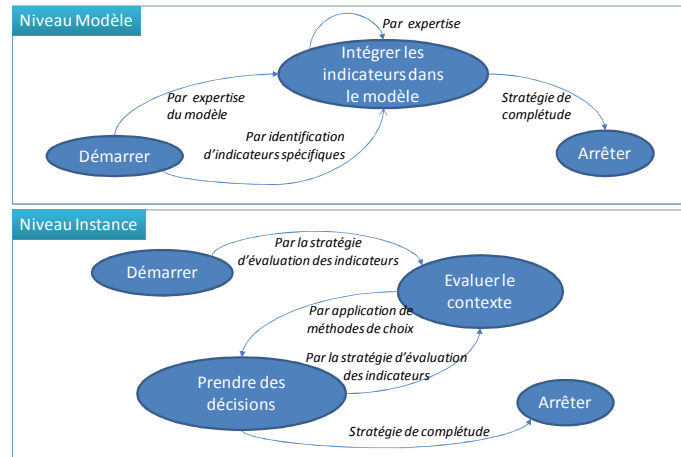


Figure 2. Processus IG

Au niveau instance, l'ingénieur de développement doit évaluer le contexte en affectant une valeur à chaque couple (indicateur, concept) (*par la stratégie d'évaluation des indicateurs*). Ensuite, l'utilisateur peut utiliser ces valeurs de caractérisation du contexte pour prendre des décisions (*par application de méthodes de choix*). Chaque décision prise peut ensuite être prise en compte par l'ingénieur pour effectuer une réévaluation de la valeur des indicateurs (*par la stratégie d'évaluation des indicateurs*) avant la prochaine prise de décision. Ce processus d'utilisation s'arrête lorsque le processus de développement ne comporte plus de points de prise de décision.

5 Application du processus de sensibilisation au modèle de processus MAP

Le modèle MAP (Rolland *et al.*, 1999) (Rolland, 2007) est un système de représentation initialement développé pour représenter le modèle de processus en termes intentionnels. La plupart des modèles n'intègrent pas réellement le concept d'intention, se concentrant sur la manière dont le processus est exécuté et externalisant dans l'intention ce que le processus est destiné à accomplir (Dietz, 2004). En revanche, les modèles de processus intentionnels se concentrent sur ce que le processus essaie d'atteindre, offrant ainsi la raison d'être du processus, à savoir pourquoi le processus est exécuté. MAP permet de spécifier les processus d'une manière flexible en mettant l'accent sur les intentions de processus et sur les différentes manières d'atteindre chacune de ces intentions. Comme le montre l'exemple ci-dessous, une carte est présentée comme un diagramme dont les nœuds sont les intentions et les arcs les stratégies. Cette carte représente le processus de l'approche Crews-L'écritaire et est décrite en détails dans (Ralyté *et al.*, 1999).

différentes. Cela permet de montrer toutes les intentions qui peuvent être atteintes après la réalisation de l'intention source.

Les sections sont à un niveau d'abstraction qui permet de capturer la variabilité du processus puisque chaque section correspond à l'application d'un service exécutable. Comme dans (Rolland *et al.*, 2007b) et (Rolland *et al.*, 2007c), notre point de vue est qu'une section représente une caractéristique visible du processus et une abstraction d'un flux cohérent d'activités exprimées de manière intentionnelle. Chaque section de la carte identifie ce que nous pouvons appeler une *variante*.

Les variantes (sections) d'une carte sont liées les unes aux autres par trois types de relations (thread, bundle et chemin) qui génèrent des multi-thread et des multi-chemins dans une carte (Prakash *et al.*, 2006).

- Un *thread* montre la possibilité d'atteindre la même intention cible, à partir d'une intention source unique, de plusieurs façons. Chacune de ces techniques est exprimée dans la carte comme une section spécifique. L'ensemble des sections participant à une telle relation est appelée un multi-thread. Cette relation exprime directement la variabilité associée à un processus multi-objectifs puisque chaque but est intégré dans les différentes stratégies permettant d'atteindre l'intention cible commune (Rolland, 2007).

- Un *chemin* établit une relation de précédence entre les sections. Pour qu'une section puisse succéder à une autre, son intention source doit être l'intention cible de la section précédente. Aucun chemin n'est recommandé a priori puisque l'ingénieur construit sa propre trajectoire suivant la situation en cours. Avec les relations de thread et de chemins, un but peut être atteint par plusieurs combinaisons de sections. Un tel ensemble est appelé un multi-chemins et peut contenir plusieurs threads. La relation multi-chemins est une autre façon de représenter la perspective pluri-objectifs du modèle de processus (Rolland, 2007).

- Un *bundle* indique la possibilité pour plusieurs sections, ayant les mêmes intentions sources et cibles, d'être mutuellement exclusives.

Une relation de raffinement montre qu'une section de la carte peut être raffinée par une autre carte. C'est un mécanisme d'abstraction par lequel un assemblage complexe de sections au niveau $i + 1$ est considéré comme une section unique au niveau i . Cela introduit le concept de niveaux de la représentation du processus. Elle implique que chaque carte peut être représentée comme une hiérarchie de cartes.

Chaque section est reliée à un service exécutable. Il fournit un moyen opérationnel de réaliser l'intention. Il implique la transformation du produit en cours d'élaboration par l'exécution d'un service (qui peut être une directive, un workflow, un algorithme, etc.).

Il existe deux directives associées à chaque section. La première est la directive de sélection d'intention (DSI). Elle définit l'ensemble des intentions qui peuvent être obtenues à l'étape suivante (lorsque la section aura été réalisée) et propose des arguments de sélection. La seconde est appelée directive de sélection de stratégie

(DSS). Elle complète la première car elle détermine toutes les stratégies pouvant être utilisées pour atteindre le but visé (identifiée avec la directive précédente) ainsi que les arguments permettant à l'ingénieur de faire son choix.

Première phase : Au niveau modèle

La première étape consiste en l'intégration du concept d'indicateur dans le modèle de processus d'origine. Le concept d'indicateur peut se rattacher à quatre concepts du modèle MAP : l'Intention, la Stratégie, la Section et la Carte elle-même, ainsi qu'il est indiqué dans la figure suivante.

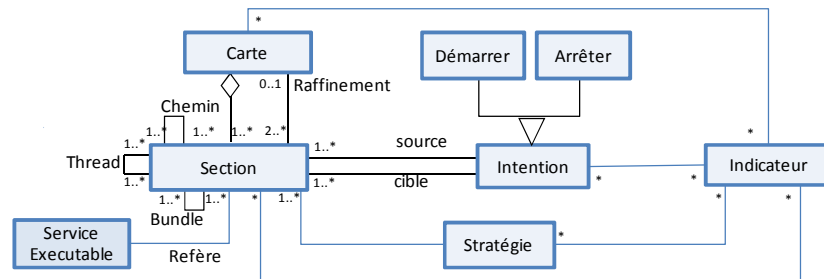


Figure 5. Le modèle de processus MAP sensible au contexte.

Deux concepts ont été supprimés pendant l'intégration de celui d'indicateur : ceux des Directives de sélection d'intention et de Directive de sélection de stratégie. En effet, ces directives, écrites en langage naturel, pourront se traduire par des valeurs d'indicateurs appliqués aux concepts d'Intention et de Stratégie.

La suite de cette étape est de définir les couples (indicateur, concept). Nous allons illustrer notre propos avec l'indicateur 'Degré d'expertise' ((Deneckere et al., 2010) propose une illustration plus complète de l'application d'indicateurs au modèle MAP). Cet indicateur va essentiellement se définir pour les stratégies. En effet, le fait d'effectuer une tâche d'une certaine manière va correspondre à un niveau d'expertise minimal. Par relation, la Section et la Carte également vont pouvoir être définies pour un niveau d'expertise particulier. Les couples obtenus sont donc les suivants : (Degré d'expertise, Stratégie), (Degré d'expertise, Section), (Degré d'expertise, Carte).

La deuxième étape permet d'identifier les indicateurs spécifiques au modèle MAP. Nous pouvons identifier quatre nouveaux indicateurs pour la typologie applicable aux cartes représentant des indicateurs dynamiques. Ces indicateurs, évalués à la volée pendant l'exécution du processus, tiennent compte de la situation en évolution constante du produit en cours de construction. Par exemple, si l'exécution d'une section ne réalise pas complètement l'intention cible, il sera alors nécessaire d'exécuter un cycle sur cette intention, afin de la réaliser plus complètement (Indicateur *Etat de l'intention*). Des indicateurs fournissent également

une évaluation de la complétude de la section. Suite à la situation du produit en construction, une section peut devoir être exécutée à plusieurs reprises afin de réaliser complètement la directive (Indicateur *Réalisation de la directive*).

La fin de la deuxième étape permet de construire les couples. Nous illustrerons cette étape avec l'indicateur spécifique *Etat de l'intention*. Cet indicateur s'applique au concept de Section (la section peut être exécutée si l'intention cible n'est pas complètement réalisée). Le couple obtenu est donc (Etat de l'intention, Section).

Deuxième phase : Au niveau instance

La première étape consiste à évaluer le contexte à l'aide des indicateurs génériques et spécifiques. L'indicateur générique choisi pour illustrer le processus est celui du Degré d'expertise. Cet indicateur s'applique tout d'abord au concept de Stratégie. Par relation, la même valeur pourra être affectée à la Section utilisant cette stratégie, sachant que le niveau d'expertise se définit avec les valeurs Faible, Moyen et Elevé. En ce qui concerne l'indicateur Etat de l'intention, le concept impacté est la Section et les valeurs sont Réalisée ou Faible réalisation. Dans la carte de l'exemple (Figure 3), nous pouvons leur affecter les valeurs indiquées dans la Table 1.

<i>Stratégie</i>	<i>Degré d'expertise</i>	<i>Section</i>	<i>Degré d'expertise</i>	<i>Etat de l'intention</i>
Identification du but initial	Faible	S1	Faible	Réalisée
Dirigée par les formulaires	Moyen	S2	Elevé	Faible réalisation
Dirigée par la structure du but	Elevé	S3	Moyen	Faible réalisation
Linguistique	Moyen	S4	Moyen	Faible réalisation
Ecriture libre	Faible	S5	Elevé	Réalisée
Manuellement	Elevé	S6	Faible	Réalisée
Outillée	Moyen	S7	Moyen	Réalisée
Découverte alternative	Faible	S8	Faible	Réalisée
Découverte par composition	Faible	S9	Faible	Réalisée
Découverte par raffinement	Faible	S10	Faible	Réalisée
Complétude	Faible	S11	Faible	Réalisée
		S12	Faible	Réalisée

Table 1. Valeurs des indicateurs de la carte d'exemple.

La deuxième étape correspond à l'utilisation de ce contexte pour la prise de décision. Lorsque l'utilisateur navigue dans la carte, ses choix se situent au niveau de l'enchaînement des différents services exécutables (choix d'une section). Pour l'illustration de notre propos, imaginons un utilisateur ayant un niveau d'expertise faible et venant de réaliser complètement l'intention 'Identifier un objectif'. Il doit maintenant choisir sa prochaine étape parmi cinq possibilités (sections S2, S3, S4, S5 et S6). L'utilisation de l'indicateur 'Etat de l'intention' lui permet de sélectionner soit l'exécution d'une section permettant de réaliser plus complètement la même intention (S2, S3, S4) soit celle d'une section allant à l'intention 'Ecrire un scénario' (S5, S6). Comme il a déjà réalisé complètement son intention 'Identifier un but', il

choisit la deuxième possibilité et restreint donc son choix à deux sections (S5, S6). L'utilisation de l'indicateur 'Degré d'expertise' lui permet d'éliminer S5 puisqu'il lui faudrait un plus haut niveau d'expertise. Les indicateurs lui ont donc permis de sélectionner S6 pour continuer sa navigation. Des techniques plus pointues peuvent être appliquées pour faire la sélection d'une section dans une carte en se basant sur les indicateurs, comme l'utilisation des algorithmes de graphes (Deneckere *et al.*, 2009) ou l'application de méthodes multi-critères (Deneckere *et al.*, 2010).

6 Discussion : Vers une notion de ligne de processus

La variabilité croissante en génie logiciel a conduit à la notion de *ligne de produits*. L'ingénierie des lignes de produits correspond au développement d'applications logicielles utilisant des plateformes et de la personnalisation de masse, ce qui signifie que les points communs et les différences des applications de la ligne de produit doivent être modélisés de manière générique (Pohl *et al.*, 2005). De la même manière qu'un Produit peut être perçu comme une ligne de Produit, c'est-à-dire une personnalisation spécifique d'une famille de produit, un Processus peut également être considéré comme une *ligne de Processus*, donc une personnalisation spécifique d'une famille de Processus (Rolland *et al.*, 2007c). Dans notre exemple, une carte spécifique représente alors la famille de processus et les indicateurs sont utilisés pour l'adapter au projet en cours, permettant d'obtenir par là même une ligne de processus particulière au projet. Cette capacité de dériver une configuration de processus à partir des caractéristiques communes d'une manière reproductible est basée sur le haut niveau de variabilité des modèles intentionnels et cette configuration pendant l'exécution augmente la sensibilité au contexte des processus.

7 Conclusion

Nous avons montré dans ce travail que la variabilité des processus téléologiques peut être augmentée par la sensibilisation au contexte. Nous avons défini le contexte comme une composition d'indicateurs représentant les spécificités du projet. Ces indicateurs peuvent être à la fois génériques (applicables à tout processus téléologique) ou spécifiques (concernant des concepts particuliers au type de processus utilisé). Nous avons également proposé un processus d'utilisation de ces indicateurs à deux niveaux (modèle et instance). Nous avons illustré notre propos par l'intégration des indicateurs dans le modèle de processus intentionnel MAP.

Nos travaux futurs concernent la définition d'une typologie plus complète pour obtenir une caractérisation du contexte plus performante. Les stratégies d'utilisation de ces indicateurs doivent également être définies et décrites plus en détails. De plus, nous souhaitons élaborer une représentation plus agréable des indicateurs, par des moyens graphiques permettant de mieux visualiser les critères entrant en compte dans les différents points de variation des processus.

Bibliographie

- (Abrahamsson, 2002) Abrahamsson, P., Salo, O., Ronkainen, J., and Warsta, J., *Agile software development methods: Review and analysis*. VTT Publications, 2002, Espoo.
- (Amenise et al., 1993) Armenise, P., Bandinelli, S., Ghezzi, C., Morzenti, A., "A survey and assessment of software process representation formalisms", *Int. Journal of Software Engineering and Knowledge Engineering*, Vol. 3, No. 3, 1993.
- (America et al., 2003) America P., Rommes E., Obbink J. H., "Multi-view variation modeling for scenario analysis". In *PFE'03*, 2003, p. 44–65
- (Bachmann et al., 2001) Bachmann F., Bass L., "Managing variability in software architectures". In *the 2001 Symposium on Software Reusability (SSR '01)*, 2001, ACM Press, p. 126–132.
- (Beck, 2005) Beck, K., *Extreme programming eXplained: embrace change*, 2nd ed. The XP Series. 2005, Boston, MA, USA, Addison Wesley.
- (Bessai et al., 2008) K. Bessai, B. Claudepierre, O. Saidani and S. Nurcan, "Context-aware business process evaluation and redesign", *Proceedings of BPMDS'08*, 2008.
- (Boehm, 1988) Boehm, B., "A Spiral model of software development and enhancement". *IEEE Computer*, 1988, 21, 5, p. 61–72.
- (Bosh et al., 2001) Bosch J. et al., "Variability issues in Software Product lines", *4th international workshop on Product Family engineering (PEE-4)*, 2001, Bilbao, Spain
- (Bradley et al., 2005) Bradley N. A., Dunlop M. D., "Toward a multidisciplinary model of context to support context-aware computing", *Human-Computer interaction*, 2005, Lawrence Erlbaum Associates.
- (Bunt, 1997) Bunt H., "Context and dialogue control", *Proceedings of CONTEXT'97*, 1997.
- (Cayla, 2008) Cayla D., "Organizational Learning: A Process Between Equilibrium and Evolution", *Journal of Economic Issues*, 2008, 42, 2, p.553-560.
- (Coutaz et al., 2002) Coutaz J., Rey G., "Recovering foundations for a theory of contextors", *4th ICCADUI*, 2002, Valenciennes, France.
- (Deneckere et al., 2009) Deneckere R., Kornysheva E., Rolland C., "Enhancing the Guidance of the Intentional Model MAP: Graph Theory Application", *RCIS'09*, 2009, Fes, Morocco.
- (Denecker et al., 2010) Deneckere R., Kornysheva E., "Process line configuration : an indicator-based Guidance of the Intentional Model MAP", *EMMSAD'10*, 2010.
- (Dey et al., 2001) Dey, A., Abowd, G., Salber, D., "A conceptual framework and toolkit for supporting the rapid prototyping of context-aware applications". *Human-computer Interaction*, 16 2-4 (special issue on context-aware computing), 2001, p 97–166.
- (Dietz, 2004) Dietz J.L.G., "Basic Notions Regarding Business Processes and Supporting Information Systems", *Proceedings of BPMDS'04*, 2004, Latvia, Riga, p.160-168.
- (Dowson et al., 1993) Dowson, M., "Software Process Themes and Issues", *IEEE Conf. on the Software Process*, 1993.
- (Dowson, 1988) Dowson, M., "Iteration in the Software Process", *Proc 9th Int. Conf. on Software Engineering*, 1988.
- (Drury et al., 2008) Drury J. L., Scott S. D., "Awareness in unmanned aerial vehicle operations", *International C2 journal*, Geoffrey N. Hone, 2:1, 2008.
- (Grosz et al., 1997) Grosz, G., Rolland, C., Schwer, S., Souveyet, C., Plihon, V., Si-Said, S., Ben Achour, C., Gnaho, C., "Modelling and Engineering the Requirements Engineering Process: An Overview of the Nature Approach", *Requirements Engineering Journal*, 2, 1997, p. 115-131.
- (Halmans et al., 2003) Halmans G., Pohl K., "Communicating the variability of a software-product family to customers", *Software and System Modeling*, 2(1), 2003, p.15–36

- (Halmans, 2003) Halmans J., "Communicating the variability of a software product family to customers", *Software and system modeling*, 2003, Springer-Verlag.
- (Jarke et al., 1994) Jarke, M., Pohl, K., Rolland, C., Schmitt, J. R., "Experienced-Based Method Evaluation and Improvement: A Process Modeling Approach", *CRIS series: Method and associated Tools for the Information Systems Life Cycle*, 1994, North Holland.
- (Kirsh Pinheiro et al., 2008) Kirsch Pinheiro M., Vanrompay Y., Berbers Y., "Context-aware service selection using graph matching", *Proceedings of ECOWS 2008*, vol. 411, 2008.
- (Kornysheva et al., 2007) Kornysheva E., Deneckère R., Salinesi C., « Method chunks selection by multicriteria techniques: an extension of the assembly-based approach », *Proc. of the International IFIP WG8.1 Conference ME 07*, 2007, Springer, Geneva, Switzerland.
- (Kornysheva et al., 2010) Kornysheva E., Deneckere R., Claudepierre B., "Contextualization of methods components", *RCIS'10*, Nice, France, 2010.
- (Kuloor et al., 2003) Kuloor C., Eberlein A., "Aspect-oriented requirements engineering for software product lines", In *ECB'03*, 2003, p. 98–107.
- (Liaskos et al., 2007) Liaskos S., Jiang L., Lapouchnian A., Wang Y., Yu Y., do Prado Leite J., Mylopoulos J., "Exploring the Dimensions of Variability: a Requirements Engineering Perspective". In *VAMOS'07*, 2007, Limerick, Ireland.
- (Loucopoulos et al., 1995) Loucopoulos, P., Kavakli, E., "Enterprise Modelling and the Teleological Approach to Requirements Engineering", *International Journal of Intelligent and Cooperative Information Systems*, 1995.
- (Malcom, 1967) Malcolm, N., "Explaining Behavior", *The Philosophical Review*, 76, 1, 1967.
- (Mirbel et al., 2006) Mirbel I., Ralyté J., "Situational method engineering: combining assembly-based and roadmap-driven approaches", *Requirements Engineering*, 11(1), 2006, pp. 58–78.
- (Ngo-The et al., 2005) Ngo-The, A., Ruhe, G., "Decision Support in Requirements Engineering", In *Engineering and Managing Software Requirements*, Ed. By A. Aurum and C. Wohlin, 2005.
- (Nurcan, 2004) Nurcan S., "Business Process Modeling for developing Process Oriented IT Systems", *IRMA'04*, 2004, New Orleans, USA.
- (Pohl et al., 2005) Pohl K., Böckle G., van der Linden F., "Software product line engineering: foundations, principles and techniques", 2005, Springer, Berlin Heidelberg New York.
- (Prakash et al., 2006) Prakash N., Rolland C., "Systems Design for requirements expressed as a map", *Proc. of the conference IRMA 06*, 2006, Washington DC.
- (Ralph, 2008) Ralph, P., Wand, Y., "A Teleological Process Theory of Software Development", *JAIS Theory Development Workshop*, 8, 23, 2008.
- (Rey et al., 2002) Rey, G., Coutaz, J., « Le Contexteur : une abstraction logicielle pour la réalisation de systèmes interactifs sensibles au contexte », *IHM'02*, 2002, p. 105-112.
- (Ralyté et al., 1999) Ralyté, J., Rolland, C., Plihon, V., "Method Enhancement with Scenario Based Techniques". *CAISE'99*, Springer-Verlag, Heidelberg, Germany, 1999, p. 103-118.
- (Rolland, 1998) Rolland, C., "A Comprehensive View of Process Engineering", *CAISE*, Springer-Verlag, 1998.
- (Rolland, 2007) Rolland C., "Capturing system intentionality with maps", in the book *Conceptual Modelling in Information Systems Engineering*, 2007.
- (Rolland et al., 1999) Rolland C., Prakash N., Benjamin A., "A Multi-Model View of Process Modelling". *Requirements Engineering*. Vol. 4, N° 4. 1999, Springer-Verlag London Ltd.
- (Rolland et al., 2000) Rolland, C., Nurcan, S., Grosz G., "A Decision Making Pattern for Guiding the Enterprise Knowledge Development Process", *Journal of Information and Software Technology*, Elsevier, 42, 2000, p. 313 - 331
- (Rolland et al., 2007b) Rolland C., Prakash N., "On the Adequate Modeling of Business Process Families", *BPMD'07*, 2007, Trondheim, Norway.

- (Rolland et al., 2007c) Rolland C., Prakash N., Kaabi R. S., “Variability in Business Process Families”, *Information Resources Management Association (IRMA)*, 2007.
- (Rosemann et al., 2006) Rosemann M., Recker J., “Context-aware process design: exploring the extrinsic drivers for process flexibility”, *CAISE'06*, workshops and doctoral consortium, 2006, Luxembourg, Namur University Press., p 149-158.
- (Rosen et al., 2008) Rosen M. A., Fiore S. M., Salas E., Letsky M., Warner N., “Tightly coupling cognition: understanding how communication and awareness drive coordination in teams”, *International C2 journal*, 2:1, 2008.
- (Roy, 2005) Roy, B., “Paradigms and challenges, Book chapter, In Multiple Criteria Decision Analysis - State of the Art Survey”, 2005, Springer. editor(s) J. Figueira, S. Greco, M. Ehrgott.
- (Royce, 1970) Royce, W. W., “Managing the development of large software systems: concepts and techniques”, In *Proceedings of Wescon*, 1970.
- (Ruhe, 2003) Ruhe, G., “Software Engineering Decision Support – Methodology and Applications”. In: *Innovations in Decision Support Systems*, Ed. by Tonfoni and Jain, International Series on Advanced Intelligence, Vol. 3, 2003, p. 143-174.
- (Salber et al., 1998) Salber, D., Dey, K. A., Abowd, D. G., “Ubiquitous Computing : Defining an HCI research agenda for an emerging interaction paradigm”. In Georgia Tech GVU technical report, 1998.
- (Salifu, 2007) Salifu, M., Yu, Y., Nuseibeh, B., “Specifying monitoring and switching problems in context”. *Proc. 15th Int. Conference on Requirements Engineering RE'07*, 2007
- (Schilit et al., 1994) Schilit B., Adams N., Want R., “Context-aware computing applications”, *IEEE WMCSA'94*, Santa Cruz, CA, US, 1994, p. 89-101.
- (Svanberg, 2001) Svanberg, “On the notion of variability in software product lines”, *working IEEE/IFIP conference on software architecture*, 2001.
- (Taylor, 1964) Taylor, Ch., “The Explanation of Behaviour”, 1964, London, Routledge.
- (Van de Ven, 1995) Van de Ven A.H., Poole, M.S., “Explaining Development and Change in Organizations”, *The Academy of Management Review*, Vol. 20, No. 3, 1995, p. 510-540.
- (Van Gurp, 2000) Van Gurp J., “Variability in Software Systems, the key to Software Reuse”, Licentiate Thesis, 2000, University of Groningen, Sweden.
- (Van Lamsweerde, 2009). van Lamsweerde A., “Reasoning About Alternative Requirements Options”, *Conceptual Modeling: Foundations and Applications*, 2009, p. 380-397
- (Van Slooten et al., 1996) Van Slooten K., Hodes B., “Characterising IS development projects”, *IFIP WG8.1 Conference on Method Engineering*, 1996.
- (Veblen, 1898) Veblen, Th., “Why is Economics not an Evolutionary Science?” *The Quarterly Journal of Economics*, 12, 4, 1898, p. 373-397.

Annexe : Plus d'information sur les auteurs

Pour plus d'informations sur les auteurs, merci de scanner les codes suivants.



Rébecca Deneckère

Elena Kornyshova



Pour scanner, vous pouvez télécharger le logiciel suivant sur votre mobile: http://www.lynkware.com/support_devices.php